Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Pino Zidarić

Instalacija i održavanje računalne učionice

Diplomski rad

Mentor: v. pred. dr. sc. Vedran Miletić

Rijeka, srpanj 2019.

Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Pino Zidarić

Instalacija i održavanje računalne učionice

Diplomski rad

Mentor: v. pred. dr. sc. Vedran Miletić

Rijeka, srpanj 2019.

Abstract

Modern computer classrooms at colleges and schools are seldom naively installed and maintained by a computer, but it is often a matter of central management by installing software on each particular computer. The task is to design the software stack and the settings of the modern classroom Department of Informatics based on the Linux operating system. For ease of use, it is a single-boot system. In practice, this is usually not the case, but as Linux and Windows on computers in the Department of Informatics are often placed on different physical disks, this is a good approximation. The hardware used is relatively new, less than 5 years old, x86-64 machines from Intel (i5, rarely i7) or AMD (Ryzen 5, maybe Ryzen 7).

Keywords

computer classroom, Linux, Ubuntu, VPN, Epoptes, Ansible, apt-catcher-ng, firewall

Sažetak

U današnje vrijeme suvremene računalne učionice na fakultetima i u školama najčešće se instaliraju i održavaju putem centralnoga upravljanja instalacijom softvera na svakome pojedinom računalu.

Zadatak ovoga diplomskog rada bio je osmisliti softverski stog i postavke suvremene učionice Odjela za informatiku Sveučilišta u Rijeci zasnovane na operacijskom sustavu Linux. Radi jednostavnosti same izvedbe, može se uzeti da je riječ o single-boot sustavu, što u praksi najčešće nije slučaj, ali s obzirom na to da se Linux i Windows na stvarnim računalima na Odjelu često stavljaju na različite fizičke diskove, to je dovoljno dobra aproksimacija. Hardver koji se koristi je relativno nov, ne stariji od pet godina, x86-64 računala od Intela (i5, rjeđe i7) ili AMD-a (Ryzen 5, možda Ryzen 7).

Ključne riječi

računalna učionica, Linux, Ubuntu, VPN, Epoptes, Ansible, apt-catcher-ng, vatrozid

Sadržaj

1. Uvod i motivacija

U današnje vrijeme suvremene računalne učionice na fakultetima i u školama najčešće se instaliraju i održavaju putem centralnoga upravljanja instalacijom softvera na svakome pojedinom računalu.

Zadatak ovoga diplomskog rada bio je osmisliti softverski stog i postavke suvremene učionice Odjela za informatiku Sveučilišta u Rijeci (dalje u tekstu: Odjel) zasnovane na operacijskome sustavu Linux. Radi jednostavnosti same izvedbe, može se uzeti da je riječ o single-boot sustavu, što u praksi to najčešće nije slučaj, ali s obzirom na da se Linux i Windows na stvarnim računalima na Odjelu često stavljaju na različite fizičke diskove, to je dovoljno dobra aproksimacija. Hardver koji se koristi je relativno nov, ne stariji od pet godina, x86-64 računala od Intela (i5, rjeđe i7) ili AMD-a (Ryzen 5, možda Ryzen 7).

Zahtjevi za učionicom su sljedeći: temelj je zadnji Ubuntu LTS (u trenutku zadavanja zadatka: 18.04) ili stabilni Debian (u trenutku zadavanja zadatka: 9). Zatim je potrebno na svakome računalu stvoriti korisnički račun informatičke podrške koji ima administratorski pristup (npr. sic), korisnički račun nastavnika koji ima administratorski pristup (npr. predavac) i korisnički račun studenta koji nema administratorski pristup (npr. student). Potom je potrebno omogućiti pristup nastavničkomu računalu putem virtualne privatne mreže (dalje u tekstu: VPN), uz pretpostavku da negdje postoji VPN poslužitelj na javnoj adresi koji se može koristiti, a osim toga VPN-om mora se omogućiti komunikacija između učionica s nastavničkoga računala u jednoj učionici na nastavničko računalo u drugoj učionici putem jednoga od alata OpenVPN (alternativno, mogli su biti iskorišteni alati SoftEther i Wireguard). Sljedeći korak je instalacija softvera za nadzor studentskoga rada, pri čemu je poslužiteljska strana na nastavničkome računalu, a klijentska strana na studentskime (alat Epoptes, a alternativno se mogao koristiti alat Veyon). Nakon toga postavljamo Apt-Cacher-NG (poslužiteljska strana na nastavničkom računalu, studentska računala ga koriste). Slijedi Ansible na nastavničkome računalu kako bi se moglo upravljati instalacijom softvera na studentskim računalima s jednog mjesta. Slijedi izrada skripte koja kod poziva s nastavničkoga računala na svim studentskim računalima putem vatrozida zabranjuje pristup svim stranicama na internetu osim Merlina. Također, potrebno je napraviti skriptu koja kod poziva s nastavničkoga računala na svim studentskim računalima odjavljuje studentskoga korisnika ako je prijavljen i zatim briše sve datoteke unutar kućnoga direktorija toga korisnika pa kopira sadržaj direktorija `/etc/skel` na njihovo mjesto.

2. Računalna učionica

Suvremena računalna učionica Odjela zasnovana je na operacijskome sustavu Linux -- u ovome radu korišten je stabilan Ubuntu 18.04 LTS Bionic Beaver [1]. Radi jednostavnosti same izvedbe, uzeto je u obzir da je riječ o single-boot sustavu. Hardver koji se koristi je relativno nov, ne stariji od pet godina, x86-64 računala od Intela (i5, rjeđe i7) ili AMD-a (Ryzen 5, možda Ryzen 7).

2.1. Pristup računalu

Pristup računalu moguć je putem SSH-a. Najprije je potrebno nadograditi postojeći sustav, a zatim instalirati OpenSSH [2].

```
$ sudo apt update - y
$ sudo apt install openssh-server -y
```

Dopustiti udaljeni pristup putem alata OpenSSH. Provjeriti status usluge `ssh` i ponovno pokrenuti po potrebi.

```
$ sudo systemctl enable ssh
$ sudo systemctl status ssh
$ sudo systemctl restart ssh
```

Za računalnu učionicu potrebno je svakomu računalu stvoriti korisnički račun informatičke podrške koji ima administratorski pristup (npr. podrska/sic), korisnički račun nastavnika koji ima administratorski pristup (npr. predavac) i korisnički račun studenta koji nema administratorski pristup (npr. student). Najprije je potrebno se prijaviti kao korijenski korisnik sljedećom naredbom.

```
$ sudo -i
```

Pošto smo se prijavili kao korijenski korisnik, dodaje se novoga korisnika `predavac` koji u ovom slučaju ima administratorske ovlasti, te se postave parametri koji su prikazani ispod. [4]

```
# adduser predavac
[sudo] lozinka za sic:
Adding user `predavac' ...
Adding new group `predavac' (1001) ...
Adding new user `predavac' (1001) with group `predavac' ...
Creating home directory `/home/predavac' ...
Copying files from `/etc/skel' ...
Upišite novu UNIX lozinku:
Ponovno upišite novu UNIX lozinku:
passwd: password updated successfully
Changing the user information for predavac
Enter the new value, or press ENTER for the default
Full Name []: Profesor1
Room Number []: 359
Work Phone []: 051/123-123
```

```
Home Phone []: 091 095 098
Other []:
```

Nakon što je dodan korisnik `predavac` dodaje se novog korisnika `student`, koji nema administratorske ovlasti.

```
# adduser student
Adding user `student' ...
Adding new group `student' (1002) ...
Adding new user `student' (1002) with group `student' ...
Creating home directory `/home/student' ...
Copying files from `/etc/skel' ...
Upišite novu UNIX lozinku:
Ponovno upišite novu UNIX lozinku:
passwd: password updated successfully
Changing the user information for student
Enter the new value, or press ENTER for the default
Full Name []: Student1
Room Number []: 359_1
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
```

Pri završenju istoga, prelazi se na sljedeći korak, a to je: omogućiti pristup nastavničkomu računalu putem VPN-a, uz pretpostavku da negdje postoji VPN poslužitelj na javnoj adresi koji se može koristiti, a osim toga VPN-om se mora omogućiti međusobna sigurna komunikacija između različitih učionica putem alata OpenVPN [5].

2.2. Tuneliranje

Za početak ukratko ćemo pojasniti pojam virtualne privatne mreže (engl. *virtual private network*, kratica VPN), odnosno pruža nam privatnu mrežu povrh neke javne mreže kao što je Internet. Uz pomoć VPN-a možemo slati i primati podatke preko javne mreže pritom zadržavajući konfiguraciju VPN-a, odnosno sigurnosti i funkcionalnosti koje nam pruža – osigurava nam da su podatci koje šaljemo sigurni i neizmijenjeni. Podatci su sigurni na način da se šifriraju prilikom prijenosa. [3] OpenVPN je zaseban softver klijent-server tehnologije koji je instaliran na računalima koji ga koriste, pružajući fleksibilna rješenja za zaštitu podatka i informacija. OpenVPN je naziv za projekt otvorenoga koda koji je pokrenuo James Yonan [6].

2.2.1. Poslužitelj/domaćin i klijent kod VPN-a

Nekoliko je preduvjeta koje moramo zadovoljiti za pristup poslužitelju koji pruža VPN uslugu. U prijašnjem koraku definirani su ne-korijenski korisnici s administratorskim ovlastima (naredba `sudo`) i postavljen je vatrozid. Za potrebe konfiguracije trebaju nam minimalno 3 računala:

poslužitelj (engl. *server*), klijent i računalo koje potpisuje certifikate (tzv. autoritet certifikata, engl. *certificate authority*, kraće CA) [7]. Posebno računalo je za izdavanje certifikata (tehnički je moguće koristiti poslužitelj OpenVPN ili lokalno računalo za izdavanje certifikata, no zbog sigurnosnih razloga, koji će biti kasnije objašnjeni, ne preporučuje se). Prema službenoj dokumentaciji OpenVPN-a [8], računalo za izdavanje i potpisivanje zahtjeva za certifikate potrebno je postaviti na samostalno računalo. Iz tog razloga u sljedećim koracima pretpostavlja se da imamo zasebno računalo s Ubuntu 18.04 poslužiteljem koja također ima ne-korijenskog korisnika s administratorskim privilegijama i osnovnim vatrozidom.

Napomena prije konfiguracije i postavljanja OpenVPN-a: ukoliko onemogućimo provjeru autentičnosti zaporka, može doći do poteškoća prilikom prijenosa datoteka u kasnijim koracima. Da bi se izbjeglo navedeni problem, potrebno je ponovno omogućiti provjeru autentičnosti zaporka na svakom poslužitelju ili, alternativno, generirati SSH ključeve za svaki poslužitelj i dodati javni SSH ključ poslužitelja OpenVPN u `.ssh/authorized_keys` datoteku računala za izdavanje certifikata i obrnuto. [9]

Nakon što su zadovoljeni svi preduvjeti, može se započeti postavljanje OpenVPN-a koje će biti podijeljeno u nekoliko koraka. Zbog jednostavnosti, u daljem tekstu koristit će se skraćeni nazivi za poslužitelje: poslužitelj VPN (poslužitelj preko kojeg ide VPN, dalje u tekstu IP adresa je 192.168.122.16), klijent (poslužitelj koji se spaja na server i koristi VPN, dalje u tekstu IP adresa je 192.168.122.131) i certifikat CA (poslužitelj za izdavanje certifikata, dalje u tekstu IP adresa je 192.168.122.110).

2.2.2. Instaliranje OpenVPN-a i EasyRSA-a

Prije samog početka potrebno je ažurirati i nadograditi postojeće pakete. Naredbama niže navedenim ažuriramo pakete poslužitelja i VPN s korisničkim računom koji ima administratorske ovlasti. Paket `openvpn` je dostupan na repozitorijima Ubuntu-a, stoga je moguće instalirati koristeći `apt`. OpenVPN koristi sigurnosne protokole za kontrolu prijenosa podataka (*engl. Transport Layer Security/Secure Sockets Layer*, kraće i dalje u tekstu TLS [10]/ SSL [11]), što znači da koristi kriptografske protokole koji omogućuju sigurnosnu komunikaciju putem Interneta, odnosno u ovome slučaju certifikate za šifriranje prometa između servera i klijenta.

```
$ sudo apt update
$ sudo apt upgrade -y
$ sudo apt install openvpn -y
```

Zbog sigurnosnih razloga za izdavanje pouzdanih certifikata, postavit će se jednostavno računalo za

autorizaciju certifikata (u daljnjem tekstu: certifikat CA računalo). Kako bi započeli izgradnju upravitelja certifikatima CA (*engl. certificate authority*) i infrastrukture javnog ključa (*engl. public key infrastructure*, u daljem tekstu skraćeno PKI) infrastrukture, koristimo `wget` za preuzimanje zadnje verzije EasyRSA na certifikat CA poslužitelju i OpenVPN poslužitelju. Najnoviju verziju preuzmemo sa službenih stranica projekta EasyRSA Github [12], gdje kopiramo vezu za preuzimanje datoteke koja završava s `.tgz`, a zatim zalijepimo kako je prikazano naredbom ispod.

```
$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/
EasyRSA-v3.0.6.tgz
```

Kao što je naprijed navedeno, na zasebnom poslužitelju (računalo navedeno gore kao certifikat CA) bit će privatni ključ za potpisivanje novih certifikata zbog zaštite od napada na VPN poslužitelj - ukoliko napadač pristupi privatnomu ključu, može potpisivati nove certifikate bez našega znanja dajući pristup VPN-u. Upravo zbog toga upravljanje s certifikatima je preporučljivo sa samostalnog certifikat CA poslužitelja, čime sprečavamo neovlaštene korisnike u pristupanju VPN-u. Pozicioniramo se u mapu gdje je preuzeta arhiva te zatim izdvojimo arhivirane datoteke naredbom `tar`.

```
$ cd ~
$ tar -xvf EasyRsa-v3.0.6.tgz
```

Na ovaj način uspješno je instaliran sav potreban softver na poslužitelju VPN i certifikat CA računalu. U daljnjim koracima konfigurirat će se EasyRSA i postaviti certifikat CA računalo za potpisivanje certifikata za pristup poslužitelju i klijentima usluzi VPN. [13]

2.2.2.1. Konfiguriranje EasyRSA varijabli i izgradnja upravitelja certifikatima

Unutar EasyRSA direktorija nalaze se konfiguracijske datoteke koje će se koristi za definiranje i konfiguriranje varijabla. Na računalu certifikat CA, naredbom ispod pozicionira se u direktorij EasyRSA.

```
$ cd ~/EasyRSA-v3.0.6/
```

Unutar ovoga direktorija nalazi se datoteka pod nazivom `vars.example`. Napravimo kopiju te datoteke i navedemo kopiju `vars` bez ekstenzije `.example`.

```
$ cp vars.example vars
```

Otvorimo datoteku pomoću nekih od uređivača teksta, u ovom slučaju korišten je uređivač `nano` (ukoliko nemamo instalirani uređivač, navedeno napravimo pomoću naredbe `sudo apt-get install nano -y`).

```
$ sudo nano vars
```

Unutar datoteke pronađemo zadane postavke za nove certifikate te ih promijenimo sukladno našoj potrebi. Izbrišemo komentare # te promijenimo vrijednosti prema potrebama, niže je naveden primjer što treba sadržavati datoteka `vars`. Nakon što izmijenimo varijable, datoteka se spremi i zatvori.

```
. . .
set_var EASYRSA_REQ_COUNTRY "HR"
set_var EASYRSA_REQ_PROVINCE "Primorsko-Goranska"
set_var EASYRSA_REQ_CITY "Rijeka"
set_var EASYRSA_REQ_ORG "Odjel za Informatiku"
set_var EASYRSA_REQ_EMAIL "uniri@uniri.hr"
set_var EASYRSA_REQ_OU "O-366"
. . .
```

Unutar direktorija EasyRSA nalazi se skripta `easyrsa` koju se poziva za obavljanje zadataka za izgradnju i upravljanje certifikatima. Pokretanjem skripte s opcijom `init-pki` pokrećemo infrastrukturu izgradnje javnog ključa na cartifikat CA računalu.

```
$ ./easyrsa init-pki
```

Ispis koji dobijemo:

```
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/sic/EasyRSA-v3.0.6/pki
```

Zatim ponovnim pozivom `easyrsa` skripte s opcijom `build-ca` izgradi se autorizacijski certifikat i dvije datoteke - `ca.cert` koja čini javnu stranu i privatnu stranu SSL certifikata`ca.key`. * `ca.cert` je javna datoteka certifikat CA računala koja služi za međusobnu komunikaciju i prepoznavanje između VPN poslužitelja i klijenta, zbog toga poslužitelj i svi klijenti trebaju imati kopiju ove datoteke. * `ca.key` je privatni ključ koji CA računalo koristi za potpisivanje ključeva i certifikata za VPN poslužitelja i klijente.

Ukoliko ne želimo svaki put upisivati lozinku kada radimo interakciju s certifikat CA računalom, možemo pokrenuti `build-ca` naredbu sa `nopass` opcijom.

```
$ ./easyrsa build-ca nopass
```

U ispisu dolazi poruka i upit za unos naziva certifikat CA računala. Radi jednostavnosti, pritiskom tipke `Enter` koristi se generičko ime, a ukoliko se želi specificirati naziv, može se unijeti bilo koji niz znakova za zajednički naziv certifikat CA računala.

```
Izlaz:
. . .
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
```

Nakon ovoga postavili smo upravitelja certifikatima, certifikat CA računala i spremno je za potpisivanje zahtjeva za certifikate. Prelazimo na generiranje privatnoga ključa i zahtjev za

certifikat od strane poslužitelj VPN računala te nakon toga šaljemo upit za potpisivanje certifikata ponovno na certifikat CA računalo. Prilikom postupka šifriranja, moguće je postaviti dodatne datoteke i varijable za poslužitelja, od kojih će neke biti navedene ispod.

2.2.2.2. Stvaranje certifikata, ključa i datoteka za šifriranje poslužitelja

Na računalu koje je predviđeno kao poslužitelj usluge VPN, najprije se pozicioniramo u direktorij EasyRSA.

```
$ cd EasyRSA-v3.0.6/
```

Nakon toga pokreće se skripta `easyrsa` s opcijom `init-pki`. Iako je ista naredba pokrenuta i na certifikat CA računalu, potrebno je pokrenuti je i na poslužitelju s obzirom na to da dva računala imaju odvojene `PKI` direktorije.

```
$ ./easyrsa init-pki
```

Ponovnim pozivom skripte `easyrsa` s opcijom `gen-req` te zajedničkim nazivom za poslužitelja usluge VPN i dodatnom opcijom `nopass`. Kada se radi s više računala, može se koristiti neko opisno ime radi kasnijega lakšeg upravljanja. Ukoliko odaberemo neko ime koje nije `server`, kasnije je potrebno prilagoditi nekoliko dolje navedenih uputa (npr. ukoliko odaberemo neko drugo ime prilikom kopiranja generiranih datoteka u `/etc/openvpn` direktorij, potrebno je zamijeniti ispravnim imenima, te promijeniti konfiguracijsku datoteku `server.conf` da pokazuje na ispravne `.crt` i `.key` datoteke).

```
$ ./easyrsa gen-req server nopass
```

Ovom naredbom stvoren je privatni ključ VPN poslužitelj i datoteka `server.req` - zahtjev za potpis certifikata. Ključ poslužitelja kopiramo u `/etc/openvpn/` direktorij.

```
$ sudo cp ~/EasyRSA-v3.0.6/pki/private/server.key /etc/openvpn/
```

Koristeći sigurnosnu metodu (kao što je SCP, u primjeru ispod) sigurnosno kopiramo `server.req` datoteku na certifikat CA računalo.

```
$ scp ~/EasyRSA-v3.0.6/pki/reqs/server.req korisnik@192.168.122.110:/root
```

Zatim na certifikat CA računalu unutar EasyRSA direktorija raspakiramo datoteku `server.req` pozivom skripte `easyrsa` s opcijom `import-req` te potom potpišemo zahtjev za certifikat od strane VPN poslužitelja s opcijom `sign-req`, nakon koje slijedi vrsta zahtjeva i uobičajeno ime. Vrsta zahtjeva može biti za poslužitelja ili klijenta. Za zahtjev VPN poslužitelja obavezno upotrebljava se `server` vrsta zahtjeva. Prilikom izlaza potrebno je potvrditi utipkavanjem `yes` zbog potvrde da zahtjev dolazi iz pouzdanoga izvora.

```
$ cd EasyRSA-v3.0.6/
$ ./easyrsa import-req /root/server.req server
$ ./easyrsa sign-req server server
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this
request has not been cryptographically verified. Please be sure it came
from a trusted source or that you have verified the request checksum with
the sender.

Request subject, to be signed as a server certificate for 3650 days:

subject=
commonName = server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: ``yes``
```

Ukoliko je šifriran ključ na certifikat CA računalu, bit će upit za lozinku u koraku iznad. Nakon toga potpisani certifikat ponovno prenesemo na poslužitelj pomoću sigurne metode za kopiranje `scp`. Isto tako i datoteku `ca.crt`.

```
$ scp pki/issued/server.crt korisnik@192.168.122.16:/root
$ scp pki/ca.crt korisnik@192.168.122.16:/root
```

Nadalje, prijavi se na OpenVPN poslužitelj i kopira `server.crt` i `ca.crt` datoteke u `/etc/openvpn/` direktorij.

```
$ sudo cp /root/{server.crt,ca.crt} /etc/openvpn
```

Zatim se pozicionira u direktorij EasyRSA i generira Diffie-Hellman ključ za sigurnosnu razmjenu ključeva. Ovo može potrajati nekoliko minuta, nakon čega se generira HMAC potpis u svrhu verifikacije TLS protokola. [3] [4]

```
$ cd EasyRSA-v3.0.6/
$ ./easyrsa gen-dh
$ openvpn --genkey --secret ta.key
```

Nakon toga, kopira se dvije nove datoteke u direktorij `/etc/openvpn/`.

```
$ sudo cp ~/EasyRSA-v3.0.6/ta.key /etc/openvpn/
$ sudo cp ~/EasyRSA-v3.0.6/pki/dh.pem /etc/openvpn/
```

Time su generirane sve datoteke certifikata i ključeva koje su potrebne poslužitelju. Sve je spremno za stvaranje odgovarajućih certifikata i ključeva koje će koristiti klijent računala za pristup OpenVPN poslužitelju.

2.2.2.3. Generiranje klijentskog certifikata i para ključeva

Iako se može generirati privatni ključ i zahtjev za certifikatom na klijentskom računalu, a zatim ga poslati na certifikat CA računalo koji ga treba potpisati, za potrebe ovoga diplomskog rada opisuje se proces generiranja zahtjeva za certifikatom na VPN poslužitelju. Prednost ovoga je da se može

napisati skriptu koja će automatski generirati konfiguracijske datoteke klijenta koje sadrže sve potrebne ključeve i certifikate. To omogućuje da se izbjegne prebacivanje ključeva, certifikata i konfiguracijskih datoteka klijentima i pojednostavljuje proces spajanja na `openvpn` uslugu.

Generirat će se jedan ključ klijenta i par certifikata. Ukoliko je više klijenata, možete se ponoviti ovaj postupak za svaki od njih. Međutim, tada je potrebno proslijediti jedinstvenu vrijednost imena skripti za svakoga klijenta. Nadalje kroz primjer prvi se certifikat/par ključeva naziva `client1`. Započinje se stvaranjem strukture direktorija unutar kućnog direktorija za pohranu klijentskoga certifikata i ključnih datoteka.

```
$ mkdir -p ~/client-configs/keys
```

Budući da će se spremiti parovi certifikata / ključevi klijenata i konfiguracijske datoteke u ovom direktoriju, potrebno je blokirati njegove dozvole kao sigurnosnu mjeru naredbom `chmod -R go-rwx ~/client-configs`.

Zatim se vratimo u direktorij EasyRSA i pokrenemo `easyrsa` skriptu s opcijama `gen-req` i `nopass` opcijom, zajedno sa zajedničkim nazivom klijenta.

```
$ cd ~/EasyRSA-v3.0.6/
$ ./easyrsa gen-req client1 nopass
```

Pritisne se `ENTER` za potvrdu uobičajenog imena. Zatim kopira `client1.key` datoteku u `/client-configs/keys/` direktorij koji je ranije stvoren `cp pki/private/client1.key ~/client-configs/keys/` Zatim se prenese `client1.req` datoteku na certifikat CA računalo pomoću sigurne metode kopiranja.

```
$ scp pki/reqs/client1.req korisnik@192.168.122.110:/root
```

Nakon toga ponovno se prijavi na certifikat CA računalo, pozicionira u direktorij EasyRSA i uveze zahtjev za certifikat.

```
$ ssh korisnik@192.168.122.110
$ cd EasyRSA-v3.0.6/
$ ./easyrsa import-req /root/client1.req client1
```

Potom potpiše zahtjev kao što je to napravljeno za VPN poslužitelj iznad, samo što je potrebno ovaj put upisati `client` umjesto `server`.

```
$ ./easyrsa sign-req client client1
```

Kada se pojavi upit, upiše se `yes` za potvrđivanje potpisa zahtjeva za certifikat i da je došao iz pouzdanog izvora. Opet, ukoliko je šifriran ključ CA, tražit će se lozinka.

```
Output
Type the word 'yes' to continue, or any other input to abort.
Confirm request details: ``yes``
```

Ovim je stvorena datoteka klijentskoga certifikata pod nazivom `client1.crt`. Prenese se ovu datoteku natrag na poslužitelj koristeći sigurnosnu metodu.

```
$ scp pki/issued/client1.crt korisnik@_server_ip:/root
```

SSH natrag na VPN poslužitelj i kopira se certifikat klijenta u `/client-configs/keys/` direktorij.

```
$ cp /root/client1.crt ~/client-configs/keys/
```

Zatim se kopira `ca.crt` i `ta.key` datoteke u `/client-configs/keys/` direktorij.

```
$ cp ~/EasyRSA-v3.0.6/ta.key ~/client-configs/keys/
$ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
```

Time su certifikati i ključevi poslužitelja i klijenta generirani i pohranjeni u odgovarajuće direktorije na VPN poslužitelju. Još uvijek postoji nekoliko radnji koje je potrebno izvršiti s tim datotekama, ali navedeno slijedi kasnije. Prelazimo na konfiguriranje OpenVPN-a na poslužitelju.

2.2.3. Konfiguriranje usluge OpenVPN

Nakon što su generirani certifikati i ključevi klijenta i VPN poslužitelja, započinje se s konfiguracijom usluge OpenVPN. Najprije se kopira primjer OpenVPN konfiguracijske datoteke u konfiguracijski direktorij i zatim izdvoji kao osnova za postavljanje.

```
$ sudo cp
/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

Otvori se datoteku konfiguracije VPN poslužitelja u željenome uređivaču teksta, u ovome slučaju `nano` uređivač teksta. Unutar konfiguracijske datoteke potrebno je pronaći `HMAC` sekciju tražeći `tls-auth` smjernicu te kraj toga dijela ukloniti komentar ukoliko nije unaprijed uklonjen. Komentar se uklanja tako da se ukloni `;` ispred linije `tls-auth ta.key 0` te ukoliko nije postavljeno na `0`, postaviti. Nakon te linije potrebno je dodati `key-direction` s postavljanim parametrom `0`.

```
. . .
tls-auth ta.key 0 # This file is secret
key-direction 0
. . .
```

Iza toga trebamo pronaći odjeljak o kriptografskim šiframa tako da pretražimo komentare s ključnom riječju `cipher` te postavi se `cipher AES-256-CBC` koji pruža razinu šifriranja i dobru podršku. Potom se dodaje `auth` smjernicu za odabir algoritma šifriranja. HMAC što je kod za provjeru autentičnosti poruke s ključem, u ovom slučaju je korišten SHA256 (algoritam za haširanje).

```
. . .
```

```
cipher AES-256-CBC
auth SHA256
. . .
```

Zatim je potrebno pronaći red koji sadrži `dh` smjernicu koja definira Diffie-Hellmanove parametre za razmjenu ključeva. Zbog promjena u programu EasyRSA, naziv datoteke Diffie-Hellman ključa može se razlikovati od onoga što je navedeno u datoteci konfiguracije poslužitelja primjera. Ako je potrebno, promijeni se naziv datoteke ovdje naveden uklanjanjem `2048` koji je postavljen zbog sigurnosnih razloga tako da se poravnava s ključem koji je generiran u prethodnome koraku.

```
. . .
dh dh.pem
. . .
```

Na kraju pronaći `user group` postavke i ukloniti `;` na početku svakog da se ukloni komentar. Za Linux sustave potrebno je da je uključeno `user nobody` i `user nogroup`. Svi ovi parametri i promjene koje su napravljene u `server.conf` datoteci do ovoga dijela su potrebne kako bi OpenVPN funkcionirao.

```
. . .
user nobody
group nogroup
. . .
```

U nastavku bit će navedene neke od promjena/postavka koje su izborne.

Dodatne postavke - DNS promjene kako bi sav promet bio preusmjeren putem VPN-a (sav promet između dva računala bit će prisiljen na korištenje tunela). Otvorimo konfiguracijsku datoteku `server.conf` i postavimo vrijednost `push "redirect-gateway def1 bypass-dhcp"`, ukoliko je uključen `;` maknemo ispred varijable, čime se postiže korištenje VPN-a za usmjeravanje cijeloga prometa i prosljeđivanje `DNS` postavki na klijentska računala s `push "dhcp-option DNS 208.67.222.222"` i `push "dhcp-option DNS 208.67.220.220"` Ovim je postignuto da se rekonfiguriraju DNS postavke na klijent računalima kako bi koristile VPN tunel kao zadani pristupnik (*engl.gateway*).

```
. . .
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 208.67.222.222"
```

Naredbom `sudo nano /etc/openvpn/server.conf` ponovno otvorimo datoteku.

```
. . .
push "dhcp-option DNS 208.67.220.220"
. . .
```

Konfiguracijska datoteka `server.conf` treba izgledati kao u sekciji ispod. (Možemo je npr. uređivati naredbom `sudo nano /etc/openvpn/server.conf`.)

```
port 1194
proto tcp
dev tun
ca ca.crt
cert server.crt
key server.key # This file should be kept secret
dh dh.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
keepalive 10 120
tls-auth ta.key 0 # This file is secret
key-direction 0
cipher AES-256-CBC
auth SHA256
user nobody
group nogroup
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
explicit-exit-notify 0
```

2.2.3.1. Podešavanje priključka i protokola

Prema zadanim postavkama, OpenVPN poslužitelj koristi `port 1194` i `UDP` protokol za prihvaćanje klijentskih veza. Ako je iz nekoga razloga potrebno koristiti drugi priključak zbog ograničavajućih mrežnih okruženja u kojima klijenti mogu biti, može se promijeniti `port` opcija. Ako se ne koriste usluge poslužitelja za web-sadržaj na OpenVPN poslužitelju, `port` 443 je popularan izbor jer je obično dopušten kroz pravila vatrozida (u ovome slučaju nije korišten). Ukoliko je protokol ograničen i na taj port, moguće je promijeniti varijablu `protos` sa UDP [15] na TCP [16] ukoliko je važno da bude pouzdana i garantirana isporuka podataka u kontroliranom redoslijedu od pošiljatelja prema primatelju. Ukoliko odaberemo `tcp` protokol, onda je potrebno izmijeniti i varijablu `explicit-exit-notify 0`, jer ovu smjernicu koristi samo protokol `udp`. Preporuka je ukoliko nema potrebe koristiti drugi port i protokol, najbolje je ostaviti te dvije postavke kao zadane. Ukoliko je korišteno drugo ime nekoliko koraka unazad umjesto `server` za certifikate/ključeve, potrebno je izmijeniti naziv koji je odabran prethodno (linije na koje je potrebno obratiti pažnju su: `cert NekoDrugoIme.crt` i `key NekoDrugoIme.key`). Nakon što je sve postavljeno, potrebno je spremiti sve promjene konfiguracijske datoteke.

2.2.3.2. Podešavanje konfiguracije mrežnog poslužitelja

Postoje neki aspekti mrežne konfiguracije VPN poslužitelja koje je potrebno prilagoditi tako da OpenVPN može ispravno usmjeriti promet kroz VPN. Prva od njih je IP prosljeđivanje, metoda za

određivanje gdje bi se IP promet trebao preusmjeriti [17]. To je bitno za VPN funkcionalnost koju poslužitelj pruža. Prilagodi se zadana postavka IP prosljeđivanja VPN poslužitelja izmjenom `/etc/sysctl.conf` datoteke.

```
$ sudo nano /etc/sysctl.conf
```

Unutra se potraži stavke `net.ipv4.ip_forward` i `net.ipv6.conf.all.forwarding` te se postavi vrijednost na `1`. Ukloni se znak `#` od početka retka što je komentar iz ove postavke. Po završetku potrebno je spremiti i zatvoriti datoteku `sysctl.conf`.

```
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
```

Da bi se pročitalo datoteku i prilagodilo vrijednosti za trenutnu sesiju, upiše se u komandno-linijsko sučelje `sudo sysctl -p` te se dobije izlaz.

```
$ sudo sysctl -p
. . .
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
. . .
```

Konfiguracija vatrozida potrebna je zbog blokiranja neželjenoga prometa, a za potrebe ovoga dijela potrebno je konfigurirati vatrozid za manipuliranje prometom koji dolazi na VPN poslužitelj. Da bi se omogućilo maskiranje, potrebno je postaviti iptables koncept da osigurava dinamički prijevod mrežnih adresa `NAT` za usmjeravanje klijentskih veza. Prije uređivanja konfiguracijske datoteke vatrozida, potrebno je pronaći javno mrežno sučelje uređaja. To napravimo sljedećom naredbom.

```
$ ip route | grep default
```

Rezultat je ispis sličan ovomu: `default via 192.168.122.1 dev ens3 proto dhcp metric 100` gdje je ključno za javno sučelje koje se nalazi unutar izlaza ove naredbe i slijedi riječ `dev`. U ovome primjeru rezultat prikazuje ime sučelja `ens3`. Kada je doznato ime sučelja za povezivanje zadanom rutom, otvori se datoteka `$ sudo nano /etc/ufw/before.rules` u koju se dodaje odgovarajuća konfiguracija. UFW pravila se obično dodaju pomoću ufw naredbe. Pravila navedena u `before.rules` datoteci se, međutim, čitaju i stavljaju na mjesto prije učitavanja uobičajenih UFW pravila. Pri vrhu datoteke dodaje se označene crte ispod. Time će se postaviti zadana pravila za `POSTROUTING` lanac u `nat` tablici i maskirati bilo koji promet koji dolazi putem VPN-a. U donjemu dijelu potrebno je zamijeniti sučelje koje je otkriveno prethodno `ens3` nakon `-A POSTROUTING` čime se dopušta promet od klijenta koji koristi OpenVPN-a na `ens3` od VPN poslužitelja. Po završetku spremiti i zatvoriti datoteku. [18]

```
#
# rules.before
```

```
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
# ufw-before-input
# ufw-before-output
# ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to ``ens3`` (change to the interface
you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o ens3 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

Zatim je potrebno konfigurirati vatrozid `UFW` da dopušta i proslijeđene pakete. Da bi se to omogućilo, potrebno je otvoriti datoteku `/etc/default/ufw` i unutra pronaći `DEFAULT_FORWARD_POLICY` smjernicu i promijeniti vrijednost s `DROP` na `ACCEPT`. Po završetku spremiti i zatvoriti datoteku.

```
. . .
DEFAULT_FORWARD_POLICY="ACCEPT"
. . .
```

Potom je potrebno prilagoditi sam vatrozid radi omogućivanja promet na OpenVPN. Ukoliko `port` i protokol nisu promijenjeni u `/etc/openvpn/server.conf` datoteci, potrebno je otvoriti UDP promet na port-u 1194. Ako je izmijenjen priključak i/ili protokol, zamijeni se odabrane vrijednosti. Također, ukoliko prethodno nije dozvoljen SSH port, dodaje ga se u ovome koraku. S obzirom na to da je ranije promijenjen priključak s `UDP` na `TCP`, potrebno je dodati sljedeće naredbe:

```
$ sudo ufw allow 1194/tcp
$ sudo ufw allow OpenSSH
```

Nakon dodavanja tih pravila, onemogući se i ponovno omogući vatrozid, odnosno`UFW` da ga ponovo pokrene i učita izmjene iz svih izmijenjenih datoteka naredbama `sudo ufw disable` i `sudo ufw enable`. Nakon ovoga VPN poslužitelj je konfiguriran za rukovanje OpenVPN prometom.

2.2.4. Pokretanje i omogućavanje usluge OpenVPN

OpenVPN usluga je spremna za pokretanje na VPN poslužitelju. To se radi pomoću sistemskoga uslužnog programa `systemctl`. OpenVPN poslužitelj pokreće se navodeći ime konfiguracijske datoteke kao varijablu instance nakon naziva datoteke sistemske jedinice. Poziva se konfiguracijska

datoteka za poslužitelj, stoga se dodaje `server` na kraj jedinice datoteke prilikom poziva konfiguracijske datoteke koja se nalazi u `/etc/openvpn/server.conf`. Dvaput provjerimo je li usluga uspješno pokrenuta upisivanjem parametra `start` i `status` unutar naredbe iza `systemctl`. [19]

```
$ sudo systemctl start openvpn@server
$ sudo systemctl status openvpn@server
```

Ako je sve prošlo dobro, ispis bi trebao biti sličan ovomu ispod.

```
$ sudo systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
Loaded: loaded (/lib/systemd/system/openvpn@.service; indirect; vendor
preset
Active: active (running) since Tue 2019-07-02 21:43:39 CEST; 4h 33min ago
Docs: man:openvpn(8)https://hr.wikipedia.org/wiki/UDP/
https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
https://community.openvpn.net/openvpn/wiki/HOWTO
Process: 5852 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status
/run/openvpn/%i.status 10 --cd /etc/openvpn --script-security 2 --config /
etc/openvpn/%i.conf --writepid /run/openvpn/%i.pid (code=exited, sta
Main PID: 882 (openvpn)
Status: "Initialization Sequence Completed"
Tasks: 1 (limit: 2338)
CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
└─882 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/se
server.conf --writepid /run/openvpn/server.pid
. . .
svi 02 21:43:39 server ovpn-server[882]: GID set to nogroup
svi 02 21:43:39 server ovpn-server[882]: UID set to nobody
svi 02 21:43:39 server ovpn-server[882]: MULTI: multi_init called, r=256
v=256
svi 02 21:43:39 server ovpn-server[882]: IFCONFIG POOL: base=10.8.0.4
size=62, i
svi 02 21:43:39 server ovpn-server[882]: ifconfig_pool_read(),
in='client1,10.8.
svi 02 21:43:39 server ovpn-server[882]: succeeded -> ifconfig_pool_set()
svi 02 21:43:39 server ovpn-server[882]: IFCONFIG POOL LIST
svi 02 21:43:39 server ovpn-server[882]: client1,10.8.0.4
svi 02 21:43:39 server ovpn-server[882]: MULTI: TCP INIT maxclients=1024
maxeven
svi 02 21:43:39 server ovpn-server[882]: Initialization Sequence Completed
```

Također možete provjeriti je li OpenVPN `tun0` sučelje dostupno upisivanjem naredbe `ip addr show tun0` te ukoliko je sve postavljeno, ispiše se.

```
$ ip addr show tun0
#4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UNKNOWN group default qlen 100
link/none
inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
valid_lft forever preferred_lft forever
inet6 fe80::609b:c396:5906:281/64 scope link stable-privacy
valid_lft forever preferred_lft forever
```

Nakon pokretanja usluge omogući se automatsko pokretanje prilikom pokretanju sustava naredbom

`sudo systemctl enable openvpn@server`. Ovim je OpenVPN usluga pokrenuta. Prije korištenja potrebno je stvoriti konfiguracijsku datoteku za klijentsko računalo. Unazad je prikazano kako se generiraju parovi certifikata i ključeva za klijente, a u sljedećemu koraku bit će prikazano kako stvoriti infrastrukturu kojom je olakšano automatizirano generiranje konfiguracijske datoteke klijenta.

2.2.5. Stvaranje infrastrukture konfiguracije klijenta

Stvaranje konfiguracijskih datoteka za OpenVPN klijente može biti donekle uključeno jer svaki klijent mora imati svoj vlastite postavke i svaki se mora uskladiti s postavkama navedenim u konfiguracijskoj datoteci poslužitelja. Umjesto pisanja jedne konfiguracijske datoteke koja se može koristiti samo na jednome klijentu, u ovome dijelu bit će opisan proces kreiranja infrastrukture konfiguracije klijenta za generiranje konfiguracijskih datoteka jednostavnijim putem. Najprije se stvori bazna konfiguracijska datoteka, a zatim izgradi skripta koja omogućuje kreiranje jedinstvene konfiguracijske datoteke klijenta, certifikate i ključeve prema potrebi.

Započne se stvaranjem novoga direktorija u koji će se spremati konfiguracijske datoteke klijenta unutar `client-configs` direktorija koji je ranije stvoren.

```
$ mkdir -p ~/client-configs/files
```

Zatim se kopira primjer konfiguracijske datoteke klijenta u `client-configs` direktorij zbog korištenja iste kao osnovne za daljnju konfiguraciju. [20]

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
~/client-configs/base.conf
```

Otvori se novu datoteku `base.conf` u uređivaču teksta.

```
$ sudo nano ~/client-configs/base.conf
```

Unutra, potrebno je pronaći `remote` direktivu. To upućuje klijenta na adresu OpenVPN poslužitelja - javnu IP adresu OpenVPN poslužitelja. Ukoliko je promijenjen priključak koji poslužitelj OpenVPN sluša, potrebno je promijeniti i priključak (`port`) 1194 u odabrani.

```
. . .
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 192.168.122.16 1194
. . .
```

Potrebno je provjeriti odgovara li protokol vrijednosti koja je korištena u konfiguraciji poslužitelja (u ovome slučaju promijenjeno je u protokol `tcp`).

```
proto tcp
```

Sljedeće, maknu se komentari ispred `user` i `group` naredbe uklanjanjem " ; " na početku svakoga retka.

```
. . .
# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup
. . .
```

Potrebno je pronaći smjernice/direktive koje postavljaju `ca`, `cert` i `key`. Ove smjernice postavi se u komentar jer u sljedećem koraku dodavat će se certifikate i ključeve u samu datoteku.

```
. . .
# SSL/TLS parms.
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
#ca ca.crt
#cert client.crt
#key client.key
. . .
```

Slično tome, komentira se `tls-auth` smjernicu jer ćete `ta.key` izravno dodati u konfiguracijsku datoteku klijenta.

```
. . .
# If a tls-auth key is used on the server
# then every client must also have the key.
#tls-auth ta.key 1
. . .
```

Dijelove kod `cipher` i `auth` postavi se kao u `/etc/openvpn/server.conf` datoteci.

```
. . .
cipher AES-256-CBC
auth SHA256
. . .
```

Zatim se dodaje `key-direction` smjernicu negdje u datoteku. Ova vrijednost mora biti postavljena na „`1`" da bi VPN ispravno funkcionirao na klijentskome računalu.

```
. . .
key-direction 1
. . .
```

Naposljetku dodaje se nekoliko označenih redaka. Iako se te smjernice mogu uključiti u svaku konfiguracijsku datoteku klijenta, potrebno ih je samo omogućiti za Linux klijente koji se isporučuju s `/etc/openvpn/update-resolv-conf` datotekom. Ova skripta koristi `resolvconf` uslužni program za ažuriranje DNS informacija za Linux klijente.

```
. . .
```

```
# script-security 2
# up /etc/openvpn/update-resolv-conf
# down /etc/openvpn/update-resolv-conf
. . .
```

Ukoliko klijent pokreće Linux i ima `/etc/openvpn/update-resolv-conf` datoteku, uklone se komentari iz ovih konfiguracijskih datoteka klijenta nakon što je generiran. Pri završetku spremi se i zatvori datoteku. Zatim se stvori jednostavna skripta koja će kompajlirati osnovnu konfiguraciju s relevantnim certifikatima, ključem i datotekama za šifriranje, a zatim generirati generiranu konfiguraciju u `~/client-configs/files` direktorij. Otvori novu datoteku nazvanu `make_config.sh` u `~/client-configs` direktoriju. Ova skripta skraćuje prethodni korak za konfiguraciju novih OpenVPN klijenata.

```
$ sudo nano ~/client-configs/make_config.sh
```

Unutar bash skripte dodaje se sljedeći sadržaj:

```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/client-configs/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/base.conf

cat ${BASE_CONFIG} \
<(echo -e '<ca>') \
${KEY_DIR}/ca.crt \
<(echo -e '</ca>\n<cert>') \
${KEY_DIR}/${1}.crt \
<(echo -e '</cert>\n<key>') \
${KEY_DIR}/${1}.key \
<(echo -e '</key>\n<tls-auth>') \
${KEY_DIR}/ta.key \
<(echo -e '</tls-auth>') \
> ${OUTPUT_DIR}/${1}.ovpn
```

Pri završetku sprema se i zatvara datoteku. Prije premještanja označi se ovu datoteku kao izvršnu tako da pomoću `chmod` maknemo dozvole grupi i ostalim korisnicima osim vlasnika. [14]

```
$ chmod go-rwx ~/client-configs/make_config.sh
```

Ova skripta izradi kopiju `base.conf` datoteke koja je prethodno napravljena, prikupiti sve datoteke certifikata i ključeva koji su izrađeni za klijenta, izdvojiti njihov sadržaj, dodati ih kopiji datoteke osnovne konfiguracije i izvesti sav taj sadržaj u nova konfiguracijska datoteka klijenta. To znači da se sve potrebne informacije pohranjuju na jednome mjestu umjesto da se zasebno upravlja konfiguracijom klijenta, certifikatom i ključnim datotekama. Prednost ovoga je da ukoliko je potrebno dodavati novog klijenta, može se samo pokrenuti ovu skriptu za stvaranje konfiguracijske datoteke i osigurano je da se sve važne informacije pohranjuju na jednome, lako dostupnome

mjestu. Jedino na što treba paziti je to da svaki puta kad se dodaje novi klijent, mora generirati nove ključeve i certifikate za njega prije pokretanja ove skripte i generiranja konfiguracijske datoteke.

2.2.5.1. Generiranje konfiguracija klijenta

U prethodnim koracima stvoren je certifikat klijenta i ključ po imenu `client1.crt` i `client1.key`, odnosno, u koraku gdje se generira konfiguracijsku datoteku za te vjerodajnice pomicanjem u `~/client-configs` imenik i skripte koja je napravljena na kraju prethodnoga koraka.

```
$ cd ~/client-configs
$ sudo ./make_config.sh client1
```

To će stvoriti datoteku pod nazivom `client1.ovpn` u `~/client-configs/files` imeniku.

```
$ ls ~/client-configs/files
client1.ovpn
```

Tu datoteku se prenosi na računalo čija je namjera korištenja kao klijenta. Dok će točne aplikacije koje se koriste za ostvarenje toga prijenosa ovisiti o operacijskome sustavu uređaja i osobnim postavkama, pouzdan i siguran način je korištenje `SFTP-a` (SSH protokol za prijenos datoteka [22]) ili `SCP` (engl.`Secure Copy` [23]) u pozadini. To će prenijeti klijentove VPN autentifikacijske datoteke preko šifrirane veze.

Evo primjera `SFTP` naredbe na `client1.ovpn` primjeru koji se može pokrenuti s lokalnoga računala. Datoteka `.openvpn` smješta se u kućni direktorij.

```
$ sftp predavac@192.168.122.16:client-configs/files/client1.ovpn ~/
```

2.2.5.2. Instalacija i konfiguracija klijenta

Ovaj odjeljak opisuje kako instalirati klijentski VPN profil na Linux Ubuntu računalu i slijedi primjer dan na mrežnim stranicama [8]. OpenVPN veza imat će isto ime kao što je nazvana `.ovpn` datoteka. U ovome radu, to znači da je veza imenovana `client1.ovpn`, poravnavajući se s prvom klijentskom datotekom koja je generirana.

Univerzalni način povezivanja je korištenje softvera OpenVPN. Na Ubuntu ili Debianu operacijskims sustavimaopenvpn može se instalirati kao što je to učinjeno na poslužitelju upisivanjem.

```
$ sudo apt update
$ sudo apt install openvpn
```

Konfiguriranje se odvija tako da se provjeri sadrži li trenutna distribucija `/etc/openvpn/update-resolv-conf` skriptu.

```
$ ls /etc/openvpn
update-resolv-conf
```

Zatim se uređuje konfiguracijsku datoteku klijenta OpenVPN koja je prenijeta.

```
$ sudo nano client1.ovpn
```

Ukoliko je pronađena `update-resolv-conf` datoteka, ukloni se komentar iz tri retka koja su dodana da bi se prilagodilo postavke DNS-a. Sadržaj datoteke `client1.ovpn` je sada:

```
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
```

Sada se može povezati s VPN-om tako da se `openvpn` naredbu usmjeri na konfiguracijsku datoteku klijenta.

```
$ sudo openvpn --config client1.ovpn
```

3. Nadzor rada unutar učionice

3.1. Nadzor studentskog rada alatom Epoptes

Epoptes (grčka riječ za nad-gledatelja) je alat za upravljanje i praćenje računalnih laboratorija/učionica otvorenog koda. Omogućuje emitiranje zaslona i nadzor, daljinsko izvršavanje naredbi, slanje poruka, nametanje ograničenja poput zaključavanja zaslona ili isključivanja zvuka klijenata i još mnogo toga. Može se instalirati u laboratorijima utemeljenim na Ubuntu, Debianu i OpenSUSE-u operativnim sustavima koji mogu sadržavati bilo koju kombinaciju sljedećeg: LTSP poslužitelje [24], klijente, ne LTSP poslužitelje, samostalne radne stanice, NX [25] ili XDMCP klijente [26] i slično. Epoptes je preveden na više od 40 jezika, uključujući i hrvatski te održavan je od strane profesora i studenata. [27]

3.1.1. Instalacija alata Epoptes

Epoptes alat sastoji se od povezanoga poslužiteljskog paketa nazvanog `epoptes` i klijentskoga paketa `epoptes-client`. Poslužiteljski paket instalira se na računalo, gdje će se pratiti klijente, u ovome slučaju to je nastavničko računalo kao poslužitelj, a studentska računala kao klijenti.

Postavljanje Epoptes-a za upravljanje računalnim laboratorijem Softver se instalira na poslužiteljskom računalu, u ovome slučaju to je nastavničko računalo. Najprije je potrebno nadograditi postojeće pakete s `sudo apt-get update -y` te instalirati epoptes s korisnikom koji

ima sudo ovlasti.

```
$ sudo apt-get isntall epoptes
```

Nakon instalacije potrebno je dodati korisnika u skupinu epoptes. Time je odabranim korisnicima omogućeno pokretanje grafičkoga korisničkog sučelja i upravljanje klijentima. Ime korisnik se zamijeni s određenim domaćinom koji će se koristiti alatom. [28]

```
$ sudo gpasswd -a student1 epoptes
```

Na klijentskim računalima otvaranjem terminala može se alatom ping provjeriti ukoliko klijentsko računalo pronalazi poslužitelja (ping imePoslužitelja.local). Ovom se naredbom prikaže ukoliko klijentsko računalo pronalazi poslužitelja. Nakon uspješne provjere moguće je instalirati epoptes na klijentska računala s dodatnim parametrom -y koji potvrđuje preuzimanje i instalaciju paketa.

```
$ sudo apt-get install epoptes-client -y
```

Nakon instalacije potrebno je ne klijentskome računalu izmijeniti unutar direktorija /etc/default datoteku epoptes-client tako da se upiše adresa od poslužitelja u liniji kojoj piše #SERVER=server s imenom koje koristi domaćin. Isto tako potrebno je maknuti komentar # na početku linije.

```
$ sudo nano /etc/default/epoptes-client
```

Zatim dohvatimo certifikat na klijentskome računalu.

```
$ sudo epoptes-client -c
```

Okvirni ispis koji se dobije prilikom dohvaćanja certifikata izgleda poput sekcije ispod.

```
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify error:num=18:self signed certificate
verify return:1
depth=0 C = AU, ST = Some-State, O = Internet Widgits Pty Ltd
verify return:1
    DONE
Successfully fetched certificate from IP_servera:789
```

Potom ponovno pokrećemo klijentsko računalo te se može koristiti alat na nastavničkome računalu. [29]

4. Nadogradnja i instalacija softvera u učionici

4.1. Apt-Cacher-NG

Apt-Cacher-NG je alat kojim radi na principu kao povezani posrednik (engl. `proxy`) poslužitelj - djeluje kao posrednik koji klijenti na lokalnoj mreži koriste za dijeljenje preuzetih podataka. Ovim alatom prati se stanje paketa i moguće je spajanje preuzimanja istih paketa s različitih lokacija (realnih ili simuliranih).

4.1.1. Instalacija i postavljanje alata Apt-Cacher-NG

Najprije se provjeri ukoliko su neki od postojećih paketa za nadogradnju naredbom `sudo apt update -y` te isto tako otvorimo port `3142` na vatrozidu da se dozvoli vanjski pristup. S obzirom na to da je ovaj paket dostupan unutar Ubuntu repozitorija na nastavničkome računalu, instalira se putem `apt-get`. [30]

```
$ sudo apt-get install apt-cacher-ng -y
```

Slijedi provjera ukoliko je `apt-cacher-ng` pokrenut na portu `3142`:

```
& ps -ef | grep apt-cacher-ng
apt-cac+ 807 1 1 23:42 ? 00:00:06 /usr/sbin/apt-cacher-ng SocketPath=/run/
apt-cacher-ng/socket -c /etc/apt-cacher-ng ForeGround=1
predavac 14799 14340 0 23:52 pts/0 00:00:00 grep --color=auto apt-cacher-
ng
$ netstat -an | grep "LISTEN "
```

Apt-cacher-ng se automatski pokreće nakon instalacije, međutim pokretanjem naredbe `systemctl status apt-cacher-ng` može se provjeriti stanje. Ispis koji se dobije sličan je ispisu ispod. Također, naredbom `systemctl is-enabled apt-cacher-ng` provjerimo ukoliko se pokreće prilikom ponovnoga pokretanja sustava. Ukoliko dobijemo ispis `enabled`, znači da je ispravno.

```
● apt-cacher-ng.service - Apt-Cacher NG software download proxy
Loaded: loaded (/lib/systemd/system/apt-cacher-ng.service; enabled; vendor
preset: enabled)
Active: active (running) since Sun 2019-07-07 23:42:23 CEST; 22min ago
Main PID: 807 (apt-cacher-ng)
Tasks: 4 (limit: 2338)
CGroup: /system.slice/apt-cacher-ng.service
└─807 /usr/sbin/apt-cacher-ng SocketPath=/run/apt-cacher-ng/socket -c
/etc/apt-cacher-ng ForeGround=1
. . .
```

Datoteke spremljene u pred-memoriji nalaze se u `/var/cache/apt-cacher-ng/`.

```
$ grep CacheDir /etc/apt-cacher-ng/acng.conf
CacheDir: /var/cache/apt-cacher-ng
```

Provjera log zapisa na apt-cacher-ng poslužitelju tako da se mogu vidjeti akcije klijenta. Pozicioniramo se u direktorij apt-caher-ng, naredbom: cd /var/log/apt-cacher-ng.

Na klijent računalu ili istoga domaćina konfigurira se apt da prolazi kroz apt-cacher-ng na portu 3142 prilikom preuzimanja paketa. Potrebno je promijeniti IP adresu u odgovarajućemu okruženju. Nakon toga ponovno se nadogradi sustav sudo apt-get update -q. [31]

```
$ echo "Acquire::http::Proxy \"http://192.168.122.16:3142\";" | sudo tee /
etc/apt/apt.conf.d/00proxy
```

S klijentske strane sve je transparentno i dobiva se standardni izlaz koji prikazuje glavne Ubuntu repozitorije kojima se pristupa. Provjera pred-memorije od strane apt-cacher-ng poslužitelja. Kao primjer je uzet paket curl koji se može instalirati s .ubuntu.com poslužitelja. Naredbom sudo apt-cache policy curl može se pregledati izvor repozitorija. Najprije brisanje curl i čišćenje lokalnih podataka.

```
$ sudo apt-get remove curl && sudo apt-get clean
```

Nakon toga ponovna instalacija, što će prisiliti apt-cacher-ng dohvaćanje istoga i stavljanje u pred-memoriju.

```
$ sudo apt-get install curl -y
```

Nakon toga ukoliko se potraži direktorij apt-cacher-ng/, naredbom find /var/cache/apt-cacher-ng/ | grep curl server apt-cacher-ng poslužitelja te dobije sličan ispis kao prikazan ispod.

```
/var/cache/apt-cacher-ng/uburep/pool/main/c/curl
/var/cache/apt-cacher-ng/uburep/pool/main/c/curl/libcurl4_7.58.0-
2ubuntu3.7_amd64.deb.head
/var/cache/apt-cacher-ng/uburep/pool/main/c/curl/curl_7.58.0-
2ubuntu3.7_amd64.deb
/var/cache/apt-cacher-ng/uburep/pool/main/c/curl/curl_7.58.0-
2ubuntu3.7_amd64.deb.head
/var/cache/apt-cacher-ng/uburep/pool/main/c/curl/libcurl4_7.58.0-
2ubuntu3.7_amd64.deb
```

S klijentskog računala instalira se paket curl i ponovno učita pretraživač na poslužitelju gdje se request hits poveća za +1, što dokazuje da predmemorija radi kako treba. Naredba u jednoj liniji: sudo apt-get remove curl -y && sudo apt-get clean; sudo apt-get install curl -y

Provjera, ukoliko sve radi, pogleda se u odabranome pretraživaču adresu: Vaša_IP_adresa:3142 statistiku i analizu.

```
2019-07-07 00:33 - 2019-07-08 00:33 24 (40.00%) 36 (60.00%) 60 0.84 MiB
(1.11%) 74.67 MiB (98.89%) 75.51 MiB
. . .
```

```
Transfer statistics
Since startup Recent history
Data fetched: 74 MiB 383 MiB
Data served: 75 MiB 385 MiB
Log analysis
Period Cache efficiency
Requests Data
Hits Misses Total Hits Misses Total
2019-07-07 00:33 - 2019-07-08 00:33 24 (40.00%) 36 (60.00%) 60 0.84 MiB
(1.11%) 74.67 MiB (98.89%) 75.51 MiB
2019-07-02 00:33 - 2019-07-03 00:33 2 (8.70%) 21 (91.30%) 23 0.01 MiB
(0.27%) 4.74 MiB (99.73%) 4.75 MiB
2019-07-01 00:33 - 2019-07-02 00:33 1 (2.38%) 41 (97.62%) 42 0.00 MiB
(0.00%) 110.42 MiB (100.00%) 110.42 MiB
```

4.2. Ansible

Pomoću ovog alata moguće je konfigurirati klijentska računala s računala na kojim su instalirane i konfigurirane komponente Ansible. U ovome slučaju to je nastavničko računalo s kojeg se konfiguriraju studentska računala. Komunikacija se vrši preko normalnih SSH kanala kako bi dohvatilo informacije s udaljenih računala, izdavalo naredbe i kopiralo datoteke. Prednost toga je što Ansible sustav ne zahtijeva dodatni softver za instalaciju na klijentska računala. Na taj se način pojednostavljuje administracija poslužitelja. Svaki poslužitelj koji ima dozvoljen SSH pristup dozvolom administriranja k poslužitelju može se administrirati pomoću Ansibla, bez obzira na to u kojemu se stadiju nalazi u svojemu životnom ciklusu. To znači da bilo koje računalo koje se može administrirati putem SSH-a, također se može administrirati kroz Ansible.

Ansible ima modularni pristup, što olakšava proširenje upotrebe funkcionalnosti glavnoga sustava za rješavanje specifičnih scenarija. Moduli se mogu pisati na bilo kojemu jeziku i komunicirati u standardnom JSON-u. Konfiguracijske datoteke uglavnom su napisane u `YAML` formatu podataka za serijalizaciju zbog svoje izražajne prirode i sličnosti s popularnim `markup` jezicima. Ansible može komunicirati s domaćinom ili putem alata naredbenoga retka ili njegovih konfiguracijskih skripti, poznatih kao `Playbooks`. [32]

4.2.1. Instalacija i konfiguracija Ansible

Da bismo zadovoljili osnove za korištenje Ansible poslužitelja, potrebne su dvije ili više Ubuntu 18.04 računala. Jedno računalo bit će korišteno kao Ansible poslužitelj (u ovome slučaju to je nastavničko računalo), dok ostatak kao domaćini - studentska računala. Još neki od preduvjeta koje je potrebno zadovoljiti su da bi svaki trebao imati ne-korijenskog korisnika sa `sudo` ovlastima i konfiguriran osnovni vatrozid, što je u ovom slučaju zadovoljeno. Još jedan od preduvjeta koji je

potrebno zadovoljiti je, radi lakše komunikacije, postaviti i spremiti ssh ključeve za spajanje koji se nalaze unutar zadane lokacije `~/.ssh/id_rsa`

Kada su ovi uvjeti zadovoljeni, može se početi s instalacijom Ansible softvera kao sredstva upravljanja različitim poslužiteljima, instalira se softver barem na jedno računalo. U ovome slučaju to je nastavničko računalo. [33]

Da bismo dobili najnoviju verziju programa Ansible za Ubuntu, u sustav se može dodati `PPA` projekta (osobni arhiv paketa) [21]. Prije nego se to učini, potrebno je provjeriti je li instaliran paket `software-properties-common`. Ovaj softver olakšat će upravljanje ovim i drugim neovisnim repozitorijima softvera.

```
$ sudo apt update -y
$ sudo apt install software-properties-common
```

Zatim se dodaje Ansible PPA pokretanjem sljedeće naredbe:

```
$ sudo apt-add-repository ppa:ansible/ansible
```

Pritisne se `ENTER` za prihvaćanje dodatka PPA. Zatim se još jednom osvježi indeks paketa sustava kako bi se ažurirali paketi dostupni unutar PPA paketa.

```
$ sudo apt update
```

Nakon ovog ažuriranja može se instalirati softver Ansible.

```
$ sudo apt install ansible
```

Sada poslužitelj Ansible ima sav softver potreban za administriranje domaćina. [33]

4.2.2. Konfiguriranje SSH pristupa

Ansible prvenstveno komunicira s klijentskim računalima putem SSH-a. Iako svakako ima sposobnost upravljanja SSH autentifikacijom na temelju lozinki, korištenje SSH ključeva može pomoći da se stvari održe jednostavnim. Na poslužitelju Ansible koristi se `cat` naredbu za ispis sadržaja datoteke javnoga ključa SSH-a ne-korijenskog korisnika.

```
$ cat ~/.ssh/id_rsa.pub
```

Izlaz prethodne naredbe kopira se u međuspremnik, a zatim se otvori novi terminal i poveže se s jednim od domaćina s mogućnošću primjene SSH.

```
$ ssh student1@192.168.122.131
```

Prebacite se na korijenskoga korisnika klijentskoga računala naredbom `sudo -i`.

```
$ sudo -i
```

Kao korijenski korisnik otvori se `authorized_keys` unutar direktorija `~/.ssh`.

```
$ nano ~/.ssh/authorized_keys
```

U ovu datoteku zalijepi se SSH ključ svojega Ansible poslužiteljskog korisnika, a zatim spremi datoteku i zatvori uređivač teksta. Zatim pokrene `exit` naredbu za vraćanje ne-korijenskom korisniku domaćina.

```
$ exit
```

Na kraju, budući da Ansible koristi `python` prevoditelj na kojem `/usr/bin/python` se pokreću njegovi moduli, potrebno je instalirati Python 2 na domaćinu kako bi Ansible mogao komunicirati s njim. Izvodi se sljedeće naredbe da bi se ažuriralo indeks paketa glavnoga računala i instaliralo `python` paket:

```
$ sudo apt update -y
$ sudo apt install python
```

Nakon toga ponovno se pokreće naredba `exit` da bi se zatvorilo vezu s klijentom.

```
$ exit
```

Ovaj je postupak potrebno za svaki poslužitelj (klijent računalo) koji će se kontrolirati poslužiteljem Ansible. Zatim se konfigurira Ansible poslužitelj za povezivanje s tim domaćinima koristeći Ansible-ovu `hosts` datoteku.

4.2.3. Postavljanje mogućih domaćina

Ansible prati sve poslužitelje o kojima zna kroz `hosts` datoteku. Tu datoteku je nužno postaviti prije nego što se započne komunicirati s drugim računalima. Pristupi se datoteku sa `sudo` ovlastima, kao što je prikazano naredbom ispod.

```
$ sudo nano /etc/ansible/hosts
```

Unutar datoteke su brojne primjere konfiguracija koje su komentirane (s prethodnim redom #). Ovi primjeri nisu od velike pomoći, zapravo, jer su domaćini navedeni u svakoj od njih sastavljeni. Međutim, zadrži se te primjere u datoteci radi lakše konfiguracije i implementiranja složenijih scenarija. Datoteka `hosts` je prilično fleksibilna i može se konfigurirati na nekoliko različitih načina. Sintaksa koja je upotrebljavana izgleda ovako:

```
[group_name]
alias ansible_host=your_host_ip
```

U ovome primjeru `group_name` nalazi se organizacijska oznaka koja vam omogućuje da se jednom riječju odnosi na sve poslužitelje navedene pod njom - `ucionica`, dok `alias` je samo ime koje se

odnosi na jedan određeni klijentski poslužitelj.

Dakle, za potrebe demonstracije zamišljeno je da su dva poslužitelja koje će se kontrolirati s Ansible-om. U ovome trenutku tim se poslužiteljima može pristupiti s poslužitelja Ansible upisivanjem:

```
$ ssh student1@192.168.122.131
```

Ne biste trebali biti upitani za lozinku ako je sve ispravno postavljeno. Za potrebe demonstracije, pretpostavit će se da domaćini imaju IP adrese `192.168.121.131`, `192.168.122.110`. Mi ćemo postaviti ovo gore, tako da možemo odnositi na njih pojedinačno kao`student1`i `student2` ili kao grupa s imenom `ucionica`.

Ovo je blok koji je potrebno dodati u `hosts` datoteku da bi se naprijed navedeno postiglo:

```
$ sudo nano /etc/ansible/hosts
[ucionica]
student1 ansible_host=192.168.121.131
student2 ansible_host=192.168.122.110
. . .
```

Domaćini mogu biti u više skupina i grupe mogu konfigurirati parametre za sve svoje članove. S trenutnim postavkama, ako se pokuša povezati s bilo kojim od tih domaćina pomoću opcije Ansible, naredba neće uspjeti (pod uvjetom da ne radimo to kao root korisnik). To je zato što je SSH ključ ugrađen za korijenskoga korisnika na udaljenim sustavima, a Ansible će se zadano povezati kao trenutni korisnik. Pokušaj povezivanja dobit će sljedeću pogrešku:

```
| UNREACHABLE! => {
```

```
    "changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host
    192.168.121.131 port 22: Connection timed out", "unreachable": true }
```

Na poslužitelju Ansible koristimo korisnika koji se zove `predavac`. Ansible će se pokušati povezati sa svakim domaćinom `ssh predavac@192.168.122.16`. To neće raditi ako korisnik `student1` nije na udaljenome sustavu. Može se stvoriti datoteku koja kaže svim poslužiteljima u grupi "poslužitelji" da se povežu kao korijenski korisnik. Da bi se to učinilo, kreira se direktorij u strukturi koja se zove Ansible `group_vars`. U ovoj mapi može se stvoriti datoteke oblikovane u `YAML-u` [34] za svaku grupu koju želimo konfigurirati.

```
$ sudo mkdir /etc/ansible/group_vars
$ sudo nano /etc/ansible/group_vars/servers
```

YAML datoteke počinju s "---", stoga je potrebno obratiti pažnju da se ne zaboravi taj dio.

```
$ /etc/ansible/group_vars/hosts
---
ansible_user: root
```

Po završetku spremi se i zatvori ovu datoteku.

Ukoliko se želi specificirati pojedinosti konfiguracije za svaki poslužitelj, bez obzira na grupno pridruživanje, te podatke može se staviti u datoteku na `/etc/ansible/group_vars/all`. Pojedini domaćini mogu biti konfigurirani za stvaranje datoteke nazvane po njihovom pseudonimom pod-direktorij na `/etc/ansible/host_vars`.

4.2.4. Upotreba jednostavnih i mogućih naredbi

Sada kada su domaćini postavljeni i dovoljno konfiguracijskih detalja koji omogućuju uspješno povezivanje s domaćinima, može se isprobati prvu naredbu. Sve poslužitelje koji su konfigurirani se provjeri alatom `ping` [35] upisivanjem naredbe ispod.

```
$ ansible -m ping all
```

Ping izlaz je prikazan u sekciji ispod.

```
student1 | SUCCESS => {
"ansible_facts": {
"discovered_interpreter_python": "/usr/bin/python"
},
"changed": false,
"ping": "pong"
}
student2 | SUCCESS => {
"ansible_facts": {
"discovered_interpreter_python": "/usr/bin/python"
},
"changed": false,
"ping": "pong"
}
```

Ovo je osnovni test kako bi se osiguralo da Ansible ima vezu sa svim svojim domaćinima.

`all` - označava sve domaćine. Može se jednako lako odrediti i grupu.

```
$ ansible -m ping ucionica
```

Također može se navesti pojedinačni domaćin kao `student1`.

```
$ ansible -m ping student1
```

Može se odrediti više domaćina tako da ih se odvoji s dvotočkama

```
$ ansible -m ping student1:student2
```

Dio naredbe `-m ping` je uputstvo za Ansible koristiti „ping” modul. To su u osnovi naredbe koje se mogu izvoditi na udaljenim domaćinima. Modul `ping` radi na mnogo načina, kao što je uobičajeni `ping` uslužni program u Linuxu, ali umjesto toga provjerava mogućnost povezivanja Ansible. Modul `ping` ne uzima nikakve argumente, ali s drugom naredbom može se vidjeti ispis i kako to

radi. Argumente se prosljeđuje u skriptu upisivanjem `-a`.

Modul "shell" omogućuje slanje naredbe terminala udaljenomu domaćinu i dohvaćanje rezultata. Na primjer, da bismo saznali koliko je memorije na našemu računalu `student1`, moguće je koristiti naredbu : `ansible -m shell -a 'free -m' student1`

```
student1 | CHANGED | rc=0 >>
total used free shared buff/cache Dostupno
Mem: 1992 355 885 1 752 1489
Swap: 1951 0 1951
0 0 0
```

Time je konfiguriran Ansible poslužitelj i može se putem istoga uspješno komunicirati i kontrolirati svoje domaćine.

4.3. Upravljanje studentskim računalima za potrebe ispitivanja

U ovome dijelu bit će opisano upravljanje studentskim računima u posebnim slučajevima, na primjer ukoliko želimo zabraniti pristup svim internetskim stranicama osim unaprijed određene, u ovom slučaju internet stranici Merlin. [36]

4.3.1. Ograničenje pristupa internetu korištenjem vatrozida iptables

Potrebno je napraviti skriptu koja kod poziva s nastavničkog računala na svim studentskim računalima putem vatrozida zabranjuje pristup svim stranicama na internetu osim Merlina.

Ovaj dio nije napravljen skriptno, nego pomoću alata `iptables` [37]. Uzme se ip adresa od MERLINA, pretpostavka da je statička i doda se za `port` 443 i 80, odnosno `https` i `http`, za tu adresu. Naredba za `IP` adresu stranice Merlin -- `$ dig +short moodle.srce.hr` rezultat: `161.53.3.61`.

Najprije dodamo domenu, ako ne radi, tada statičku IP adresu (DNS bi trebao `queryat`/dohvatiti domenu za IP). Alat `iptables` čita odozgo prema dolje, dakle bilo koji `rule` prije zadnjega radit će normalno, ako ga se stavi na kraj, neće raditi. [10]

Iptables za blokiranje cijeloga prometa, a zatim samo dopustiti promet s određenih IP adresa. Ova pravila vatrozida ograničavaju pristup određenim resursima na mrežnom sloju. U nastavku je primjer niza naredbi. [38]

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i lo -m comment --comment "Allow loopback connections"
-j ACCEPT
iptables -A INPUT -p icmp -m comment --comment "Allow Ping to work as
```

```
expected" -j ACCEPT
iptables -A INPUT -s 161.53.3.61 -j ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP
```

U naredbi, `iptables -A INPUT -s 161.53.3.61 -j ACCEPT` dopušten je pristup Merlin stranicama i dopuštaju sve dolazne i odlazne pakete koji su povezani s postojećim vezama. Posljednje dvije naredbe postavljaju zadanu politiku za sve `INPUT` i `FORWARD` lance da ispuste sve pakete. [39] Alternativno ovaj dio se može riješiti pomoću Playbook Ansible skripte isto tako koristeći alat `iptables` kod je prikazan ispod.

```
---
    - hosts: localhost
        remote_user: sysadmin
        become: true

        vars:
        host_name: localhost

        tasks:

        - iptables:
            chain: INPUT
            source: 192.168.1.1
            jump: ACCEPT
        become: yes

        - iptables:
            chain: OUTPUT
            destination: 192.168.1.1
            jump: ACCEPT
        become: yes

        - iptables:
            chain: INPUT
            policy: DROP
        become: yes

        - iptables:
            chain: OUTPUT
            policy: DROP
        become: yes
```

4.3.2. Bash skripta kontrola studentskih direktorija

U ovome dijelu prikazana je skripta koja poziva kod s nastavničkog računala te na svim studentskim računalima odjavljuje studentskoga korisnika ako je prijavljen i zatim briše sve datoteke unutar kućnog direktorija toga korisnika pa kopira sadržaj direktorija /etc/skel na njihovo mjesto. `Bash` skripta je zbog jednostavnosti i lakšega snalaženja podijeljena u tri sekcije.

Prva sekcija traži i izlista sve prijavljene studente te nakon toga odjavljuje prijavljene studente.

Druga sekcija nazvana `Brisanje` čisti unutar unaprijed postavljenih direktorija, odnosno datoteka –

i to radi tako da prisilno rekurzivno obriše i pregazi eventualna javljanja greški (naredba `rm -rvf`). U svakome trenutku moguće je vidjeti napredak i točno što se briše.

Treća sekcija unutar `bash` skripte postavlja i kopira sadržaj iz direktorija `etc/sekl` i postavlja na njihovo mjesto prethodno definirane datoteke.

```
#!/bin/bash

# Izlista tko je sve ulogiran, trazi samo "username" i na kraju sve osim
root
who | cut -d' ' -f1 | grep student) )

printf "\n####################\nUlogirani studenti\n####################\n\
n"

# Iteracija po polju u kojem su spremljeni korisnici
for user in "${who[@]}"
do
    echo $user
done

printf "\nOdlogiravanje svih studenta...Pozdrav s mora\n\n"
# Odlogirani studenti
for user in "${who[@]}"
do
    echo "pkill -KILL -u $user"
    echo "$user killed."
done

printf "\nRješeno!\n"

printf "\n\n############\n Brisanje \n############\n\n"

# Ciscenje user direktorija
for user in "${who[@]}"
do
    homedir=$(grep ${user} /etc/passwd | cut -d':' -f6)
    echo "Brisanje datoteka/direktorija unutar: $homedir"
    rm -rvf $homedir/*
done

printf "\n\n#############\nPostavljanje SKEL-a\n#############\n\n"

# Kopiranje datoteka iz /etc/skel direktorija

for user in "${who[@]}"
do
    cp -vr /etc/skel/. /home/$user/
    # Dohvacanje datoteka unutar /home/user direktorija i
    # promjena ownershipa za tog usera
    files=( $(ls -1) )
    for file in "${files[@]}"
        do
        chown $user:$user $file
    done
done
```

5. Zaključak

U ovome diplomskom radu imao sam ulogu sistemskoga administratora te sam, s obzirom na zadanu temu, osmislio softverski stog i postavke suvremene računalne učionice.

Za potrebe ovoga diplomskog rada sav korišteni softver je softver otvorenoga koda. Kao temelj za postavljanje učionice korišten je Ubuntu LTS 18.04. Stvaranjem korisničkoga računa informatičke podrške, nastavnika i studenata dobiveni su preduvjeti za početak daljnjih postavki računalne učionice. Zbog sigurnosti protoka podataka, korišten je OpenVPN, što nam osigurava sigurne i neizmijenjene podatke koje šaljemo. Za nadzor studentskih računala odlučio sam koristiti i konfigurirati softver otvorenoga koda Epoptes s obzirom na to da isti ima korisničko sučelje prevedeno na hrvatski jezik. Ansible je postavljen radi jednostavnije automatizacije instaliranja, postavljanja, kopiranja s nastavničkog na sva studentska računala odjednom (npr. instalacija pojedinog softvera odjednom na sva studentska računala). Ukoliko je prethodno postavljen apt-cacher-ng, naprijed opisani proces je još brži. Konačno, izradio sam skriptu koja kod poziva s nastavničkoga računala na svim studentskim računalima odjavljuje studentskoga korisnika ako je prijavljen i zatim briše sve datoteke unutar kućnoga direktorija toga korisnika pa kopira sadržaj direktorija /etc/skel na njihovo mjesto.

Osim toga, napravio sam da se prema potrebi može napraviti zabrana pristupa svim internetskim stranicama, osim onih predodređenih.

Zaključujem kako je za postavljanje suvremene računalne učionice velika prednost mogućnost korištenja softvera otvorenoga koda koji je besplatan. Samim time možemo uštedjeti s obzirom na to da nije potrebno koristiti komercijalni softver.

Jedna od poteškoća s kojom sam se susreo prilikom postavljanja je velik broj komponenti i njihova međuzavisnost. Osobno smatram kako bi samo postavljanje računalne učionice trebalo biti automatiziranije kako bi se cijeli postupak jednostavnije i lakše konfigurirao s nastavničkog na studentska računala.

6. Popis literature i izvora

[1] “Ubuntu 18.04.2 LTS (Bionic Beaver).” [Online]. Dostupno: http://releases.ubuntu.com/18.04/. [Pristupljeno: 02. srpnja 2019].

[2] “OpenSSH: Manual Pages.” [Online]. Dostupno: https://www.openssh.com/manual.html. [Pristupljeno: 02. srpnja 2019].

[3] “Tuneliranje alatom OpenVPN — CNPSLab homepage.” [Online]. Dostupno: https://lab.miletic.net/hr/nastava/materijali/openvpn-virtualna-privatna-mreza/. [Pristupljeno: 02. srpnja 2019].

[4] “How To Create a Sudo User on Ubuntu,” 07-May-2018. [Online]. Dostupno: https://linuxize.com/post/how-to-create-a-sudo-user-on-ubuntu/. [Pristupljeno: 02. srpnja 2019].

[5] “VPN Software Solutions & Services For Business,” *OpenVPN*. [Online]. Dostupno: https://openvpn.net/. [Pristupljeno: 02. srpnja 2019].

[6] “About OpenVPN,” *OpenVPN*. [Online]. Dostupno: https://openvpn.net/about/. [Pristupljeno: 02. srpnja 2019].

[7] “Certificate authority,” *Wikipedia*. 02. srpnja 2019.

[8] “How To Guide: Set Up & Configure OpenVPN client/server VPN,” *OpenVPN*. [Online]. Dostupno: https://openvpn.net/community-resources/how-to/. [Pristupljeno: 02. srpnja 2019].

[9] “SSH/OpenSSH/Configuring - Community Help Wiki.” [Online]. Dostupno: https://help.ubuntu.com/community/SSH/OpenSSH/Configuring. [Pristupljeno: 02. srpnja 2019].

[10] “TLS,” *Wikipedija*. 13-May-2015. [Pristupljeno: 10. lipnja 2019]

[11] “SSL,” *Wikipedija*. 24-Jan-2009. [Pristupljeno: 10. lipnja 2019]

[12] easy-rsa: Simple shell based CA utility. *GitHub*. OpenVPN Inc, 2019. [Online]. Dostupno: https://github.com/OpenVPN/easy-rsa [Pristupljeno: 02. svibnja 2019.]

[13] Mark Drake and Justin Ellingwood, “How To Set Up an OpenVPN Server on Ubuntu 18.04,” *DigitalOcean*. [Online]. Dostupno: https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-18-04. [Pristupljeno: 14.travnja 2019].

[14] “Ubuntu Manpage: chmod - change file mode bits.” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/trusty/man1/chmod.1.html. [Pristupljeno: 02. srpnja 2019].

[15] “User Datagram Protocol,” *Wikipedia*. 24. lipnja 2019.

[16] “Transmission Control Protocol,” *Wikipedia*. 03. srpnja 2019.

[17] “OpenVPN - Understand the routing table + How to route only the traffic to a specific ip via the VPN,” *Unix & Linux Stack Exchange*. [Online]. Dostupno: https://unix.stackexchange.com/questions/263678/openvpn-understand-the-routing-table-how-to-route-only-the-traffic-to-a-spec. [Pristupljeno: 02. srpnja 2019].

[18] “OpenVPN – forward all client traffic through tunnel using UFW,” *digital nomad*, 21-Apr-2013. [Online]. Dostupno: https://www.gaggl.com/2013/04/openvpn-forward-all-client-traffic-through-tunnel-using-ufw/. [Pristupljeno: 02. srpnja 2019].

[19] “Ubuntu Manpage: systemctl - Control the systemd system and service manager.” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/xenial/man1/systemctl.1.html. [Pristupljeno: 02. srpnja 2019].

[20] “How To Guide: Set Up & Configure OpenVPN client/server VPN,” *OpenVPN*. [Online]. Dostupno: https://openvpn.net/community-resources/how-to/. [Pristupljeno: 02. srpnja 2019].

[21] “What is PPA? Everything You Need to Know About PPA in Linux,” *https://itsfoss.com/*. [Online]. Dostupno: https://itsfoss.com/ppa-guide/. [Pristupljeno: 02. srpnja 2019].

[22] “Ubuntu Manpage: sftp — secure file transfer program.” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/bionic/en/man1/sftp.1.html. [Pristupljeno: 02. srpnja 2019].

[23] “Ubuntu Manpage: scp — secure copy (remote file copy program).” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/bionic/man1/scp.1.html. [Pristupljeno: 02. srpnja 2019].

[24] “Linux Terminal Server Project - Welcome to LTSP.org.” [Online]. Dostupno: http://www.ltsp.org/. [Pristupljeno: 02. srpnja 2019].

[25] “NX technology,” *Wikipedia*. 14-Apr-2019.

[26] “xdmcp - Ubuntu Wiki.” [Online]. Dostupno: https://wiki.ubuntu.com/xdmcp. [Pristupljeno: 02. srpnja 2019].

[27] A. Georgopoulos, “Epoptes.” [Online]. Dostupno: http://www.epoptes.org/about. [Pristupljeno: 03. srpnja 2019].

[28] A. Georgopoulos, “Installation - Epoptes.” [Online]. Dostupno: http://www.epoptes.org/installation. [Pristupljeno: 03. srpnja 2019].

[29] A. Georgopoulos, “Question #284584 : Questions : Epoptes.” [Online]. Dostupno: https://answers.launchpad.net/epoptes/+question/284584. [Pristupljeno: 03.srpnja 2019].

[30] L. Babin, “Setting up an ‘Apt-Cache’ Server Using ‘Apt-Cacher-NG’ in Ubuntu 14.04 Server.” [Online]. Dostupno: https://www.tecmint.com/apt-cache-server-in-ubuntu/. [Pristupljeno: 01. srpnja 2019].

[31] F. Lee, “Ubuntu: A centralized apt package cache using Apt-Cacher-NG – Fabian Lee : Software Architect.” [Online]. Dostupno: https://fabianlee.org/2018/02/11/ubuntu-a-centralized-apt-package-cache-using-apt-cacher-ng/. [Pristupljeno: 07. ožujka 2019].

[32] E. Heidi, “Ansible is Simple IT Automation.” [Online]. Dostupno: https://www.ansible.com. [Pristupljeno: 08. srpnja 2019].

[33] D. Mark and R.-C. Stephen, “How to Install and Configure Ansible on Ubuntu 18.04 | DigitalOcean.” [Online]. Dostupno: https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-18-04. [Pristupljeno: 02. srpnja 2019].

[34] “YAML Syntax — Ansible Documentation.” [Online]. Dostupno: https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html. [Pristupljeno: 02. srpnja 2019].

[35] “Ubuntu Manpage: ping - send ICMP ECHO_REQUEST to network hosts.” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/cosmic/man8/ping.8.html. [Pristupljeno: 02. srpnja 2019].

[36] “Merlin 2018/2019.” [Online]. Dostupno: https://moodle.srce.hr/2018-2019/. [Pristupljeno: 02. srpnja 2019].

[37] “Ubuntu Manpage: iptables/ip6tables — administration tool for IPv4/IPv6 packet filtering and NAT.” [Online]. Dostupno: http://manpages.ubuntu.com/manpages/bionic/man8/iptables.8.html. [Pristupljeno: 02. srpnja 2019].

[38] “Block Outgoing Network Access For a Single User Using Iptables,” *nixCraft*, 04-Apr-2006. [Online]. Dostupno: https://www.cyberciti.biz/tips/block-outgoing-network-access-for-a-single-user-from-my-server-using-iptables.html. [Pristupljeno: 02. srpnja 2019].

[39] “Control Network Traffic with iptables,” *Linode Guides & Tutorials*. [Online]. Dostupno: https://www.linode.com/docs/security/firewalls/control-network-traffic-with-iptables/. [Pristupljeno: 02. srpnja 2019].

Sadržaj